IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

Timestamp-Independent Motion Vector Prediction for Predictive (P) and Bidirectionally Predictive (B) Pictures

Inventor(s):
Alexandros Tourapis
Shipeng Li
Feng Wu
Gary Sullivan

ATTORNEY'S DOCKET NO. MS1-1252US

RELATED PATENT APPLICATIONS

This U.S. Non-provisional Application for Letters Patent is a continuation-in-part of co-pending U.S. Application for Letters Patent Serial No. 10/444,511, filed May 23, 2003, and titled "Spatiotemporal Prediction for Bidirectionally Predictive (B) Pictures and Motion Vector Prediction For Multi-Picture Reference Motion Compensation", which is incorporated by reference in its entirety herein.

This U.S. Non-provisional Application for Letters Patent claims the benefit of priority from, and hereby incorporates by reference the entire disclosure of, co-pending U.S. Provisional Application for Letters Patent Serial No. 60/397,187, filed July 19, 2002, and titled "Timestamp Independent Motion Vector Prediction for P and B frames with Division Elimination".

This U.S. Non-provisional Application for Letters Patent is also related to co-pending U.S. Application for Letters Patent Serial No. 10/186,284, filed June 27, 2002, and titled "Improved Video Coding Methods And Apparatuses", which is incorporated by reference in its entirety herein.

TECHNICAL FIELD

This invention relates to video coding, and more particularly to methods and apparatuses for providing improved encoding/decoding and/or prediction techniques associated with different types of video data.

BACKGROUND

There is a continuing need for improved methods and apparatuses for compressing/encoding data and decompressing/decoding data, and in particular image and video data. Improvements in coding efficiency allow for more information to be processed, transmitted and/or stored more easily by computers and other like devices. With the increasing popularity of the Internet and other like computer networks, and wireless communication systems, there is a desire to provide highly efficient coding techniques to make full use of available resources.

Rate Distortion Optimization (RDO) techniques are quite popular in video and image encoding/decoding systems since they can considerably improve encoding efficiency compared to more conventional encoding methods.

The motivation for increased coding efficiency in video coding continues and has recently led to the adoption by a standard body known as the Joint Video Team (JVT), for example, of more refined and complicated models and modes describing motion information for a given macroblock into the draft international standard known as H.264/AVC. Here, for example, it has been shown that Direct Mode, which is a mode for prediction of a region of a picture for which motion parameters for use in the prediction process are predicted in some defined way based in part on the values of data encoded for the representation of one or more of the pictures used as references, can considerably improve coding efficiency of B pictures within the draft H.264/AVC standard, by exploiting the statistical dependence that may exist between pictures.

In the draft H.264/AVC standard as it existed prior to July of 2002, however, the only statistical dependence of motion vector values that was exploited was temporal dependence which, unfortunately, implies that timestamp information for each picture must be available for use in both the encoding and decoding logic for optimal effectiveness. Furthermore, the performance of this mode tends to deteriorate as the temporal distance between video pictures increases, since temporal statistical dependence across pictures also decreases.

Problems become even greater when multiple picture referencing is enabled, as is the case of H.264/AVC codecs.

Consequently, there is continuing need for further improved methods and apparatuses that can support the latest models and modes and also possibly introduce new models and modes to take advantage of improved coding techniques.

SUMMARY

Improved methods and apparatuses are provided that can support the latest models and modes and also new models and modes to take advantage of improved coding techniques.

The above stated needs and others are met, for example, by a method for use in encoding video data. The method includes establishing a first reference picture and a second reference picture for each portion of a current video picture to be encoded within a sequence of video pictures, if possible, and dividing each current video pictures into at least one portion to be encoded or decoded. The method then includes selectively assigning at least one motion vector predictor (MVP) to a current portion of the current video picture (e.g., in which the current picture is a coded frame or field). Here, a portion may include, for example, an entire frame or field, or a slice, a macroblock, a block, a subblock, a sub-partition, or the like within the coded frame or field. The MVP may, for example, be used without alteration for the formation of a prediction for the samples in the current portion of the current video frame or field. In an alternative embodiment, the MVP may be used as a prediction to which is added an encoded motion vector difference to form the prediction for the samples in the current portion of the current video frame or field.

For example, the method may include selectively assigning one or more motion parameter to the current portion. Here, the motion parameter is associated with at least one portion of the second reference frame or field and based on at least a spatial prediction technique that uses a corresponding portion and at least one collocated portion of the second reference frame or field. In certain instances, the collocated portion is intra coded or is coded based on a different reference frame or field than the corresponding current portion. The MVP can be based on at least one motion parameter of at least one portion adjacent to the current portion within the current video frame or field, or based on at least one direction selected from a forward temporal direction and a backward temporal direction associated with at least one of the portions in the first and/or second reference frames or fields. In certain implementations, the motion parameter includes a motion vector that is set to zero when the collocated portion is substantially temporally stationary as determined from the motion parameter(s) of the collocated portion.

The method may also include encoding the current portion using a Direct Mode scheme resulting in a Direct Mode encoded current portion, encoding the current portion using a Skip Mode scheme resulting in a Skip Mode encoded current portion, and then selecting between the Direct Mode encoded current frame and the Skip Mode encoded current frame. Similarly, the method may include encoding the current portion using a Copy Mode scheme based on a spatial prediction technique to produce a Copy Mode encoded current portion, encoding the current portion using a Direct Mode scheme based on a temporal prediction technique to produce a Direct Mode encoded current portion, and then selecting between the Copy Mode encoded current portion and the Direct Mode encoded current portion. In certain implementations, the decision process may include the use of a Rate Distortion Optimization (RDO) technique or the like, and/or user inputs.

The MVP can be based on a linear prediction, such as, e.g., an averaging prediction. In some implementations the MV is based on non-linear prediction such as, e.g., a median prediction, etc. The current picture may be encoded as a B picture (a picture in which some regions are predicted from an average of two motion-compensated predictors) or a P picture (a picture in which each region has at most one motion-compensated prediction), for example and a syntax associated with the current picture configured to identify that the current frame was encoded using the MVP.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings. The same numbers are used throughout the figures to reference like components and/or features.

Fig. 1 is a block diagram depicting an exemplary computing environment that is suitable for use with certain implementations of the present invention.

Fig. 2 is a block diagram depicting an exemplary representative device that is suitable for use with certain implementations of the present invention.

Fig. 3 is an illustrative diagram depicting Direct Prediction in B picture coding, in accordance with certain exemplary implementations of the present invention.

Fig. 4 is an illustrative diagram depicting handling of collocated Intra within existing codecs wherein motion is assumed to be zero, in accordance with certain exemplary implementations of the present invention.

Fig. 5 is an illustrative diagram demonstrating that Direct Mode parameters need to be determined when the reference picture index of the collocated block in the backward reference P picture is other than zero, in accordance with certain exemplary implementations of the present invention.

Fig. 6 is an illustrative diagram showing a scene change and/or the situation wherein the collocated block is intra-coded, in accordance with certain exemplary implementations of the present invention.

Fig. 7 is an illustrative diagram depicting a scheme wherein MV_{FW} and MV_{BW} are derived from spatial prediction (e.g., Median MV of surrounding Macroblocks) and wherein if either one is not available (e.g., no predictors) then one-direction may be used, in accordance with certain exemplary implementations of the present invention.

Fig. 8 is an illustrative diagram depicting how spatial prediction may be employed to solve the problem of scene changes and/or that Direct Mode need not be restricted to being Bidirectional, in accordance with certain exemplary implementations of the present invention.

Fig. 9 is an illustrative diagram depicting Timestamp Independent SpatioTemporal Prediction for Direct Mode, in accordance with certain exemplary implementations of the present invention.

Figs. 10a-b are illustrative diagrams showing how Direct/Skip Mode decision can be performed either by an adaptive picture level RDO decision and/or by user scheme selection, in accordance with certain exemplary implementations of the present invention.

Fig. 11 is a table listing some syntax changes that can be used in header information, in accordance with certain exemplary implementations of the present invention.

Fig. 12 is an illustrative diagram depicting different frames which signal the use of a different type of prediction for their corresponding Direct (B) and Skip (P) modes. P_Z, P_T, and P_M, define for example zero, temporal and spatial prediction, and B_T, B_{SP}, define temporal and spatial prediction for Direct Mode, in accordance with certain exemplary implementations of the present invention.

Fig. 13 is a table showing modifications to modes for 8x8 blocks in B pictures/slices applicable to the H.264/AVC coding scheme, in accordance with certain exemplary implementations of the present invention.

Fig. 14 is an illustrative diagram depicting median prediction of motion vectors, in accordance with certain exemplary implementations of the present invention.

Fig. 15 is a table showing P-Picture Motion Vector prediction (e.g., Non-Skip, non-8x16, non-16x8 MBs), in accordance with certain exemplary implementations of the present invention.

Fig. 16 is an illustrative diagram depicting median prediction of motion vectors, in accordance with certain exemplary implementations of the present invention.

Fig. 17 is an illustrative diagram showing replacement of Intra subblock predictors with adjacent Inter subblock predictors, in accordance with certain exemplary implementations of the present invention.

Fig. 18 is an illustrative diagram depicting how Motion Vector Prediction of current block (C) may consider the reference frame information of the predictor macroblocks (Pr) and perform the proper adjustments (e.g., scaling of the predictors), in accordance with certain exemplary implementations of the present invention.

Fig. 19 is an illustrative diagram depicting certain exemplary predictors for 8×8 partitioning, in accordance with certain exemplary implementations of the present invention.

Fig. 20 is a table showing the relationship between previous λ and current λ, in accordance with certain exemplary implementations of the present invention.

Fig. 21 is a table showing the performance difference of exemplary proposed schemes and proposed RDO versus conventional software (i.e., H.264/AVC JM3.3), in accordance with certain exemplary implementations of the present invention.

Fig. 22 is a table showing a comparison of encoding performance for different values of λ, in accordance with certain exemplary implementations of the present invention.

Fig. 23 is an illustrative timeline showing a situation wherein reference pictures of a macroblock partition temporally precede a current picture, in accordance with certain exemplary implementations of the present invention.

DETAILED DESCRIPTION

While various methods and apparatuses are described and illustrated herein, it should be kept in mind that the techniques of the present invention are not limited to the examples described and shown in the accompanying drawings, but are also clearly adaptable to other similar existing and future video coding schemes, etc.

Before introducing such exemplary methods and apparatuses, an introduction is provided in the following section for suitable exemplary operating environments, for example, in the form of a computing device and other types of devices/appliances.

Exemplary Operational Environments:

Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer.

Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, portable communication devices, and the like.

The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Fig.1 illustrates an example of a suitable computing environment 120 on which the subsequently described systems, apparatuses and methods may be implemented. Exemplary computing environment 120 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the improved methods and systems described herein. Neither should computing environment 120 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in computing environment 120.

The improved methods and systems herein are operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable include, but are not limited to, personal computers, server computers, thin clients, thick clients, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

As shown in Fig. 1, computing environment 120 includes a general-purpose computing device in the form of a computer 130. The components of computer 130 may include one or more processors or processing units 132, a system memory 134, and a bus 136 that couples various system components including system memory 134 to processor 132.

Bus 136 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnects (PCI) bus also known as Mezzanine bus.

Computer 130 typically includes a variety of computer readable media. Such media may be any available media that is accessible by computer 130, and it includes both volatile and non-volatile media, removable and non-removable media.

In Fig. 1, system memory 134 includes computer readable media in the form of volatile memory, such as random access memory (RAM) 140, and/or non-volatile memory, such as read only memory (ROM) 138. A basic input/output system (BIOS) 142, containing the basic routines that help to transfer information between elements within computer 130, such as during start-up, is stored in ROM 138. RAM 140 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processor 132.

Computer 130 may further include other removable/non-removable, volatile/non-volatile computer storage media. For example, Fig. 1 illustrates a hard disk drive 144 for reading from and writing to a non-removable, non-volatile magnetic media (not shown and typically called a "hard drive"), a magnetic disk drive 146 for reading from and writing to a removable, non-volatile magnetic disk 148 (e.g., a "floppy disk"), and an optical disk drive 150 for reading from or writing to a removable, non-volatile optical disk 152 such as a CD-ROM/R/RW, DVD-ROM/R/RW/+R/RAM or other optical media. Hard disk drive 144, magnetic disk drive 146 and optical disk drive 150 are each connected to bus 136 by one or more interfaces 154.

The drives and associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules, and other data for computer 130. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 148 and a removable optical disk 152, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 148, optical disk 152, ROM 138, or RAM 140, including, e.g., an operating system 158, one or more application programs 160, other program modules 162, and program data 164.

The improved methods and systems described herein may be implemented within operating system 158, one or more application programs 160, other program modules 162, and/or program data 164.

A user may provide commands and information into computer 130 through input devices such as keyboard 166 and pointing device 168 (such as a "mouse"). Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, serial port, scanner, camera, etc. These and other input devices are connected to the processing unit 132 through a user input interface 170 that is coupled to bus 136, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB).

A monitor 172 or other type of display device is also connected to bus 136 via an interface, such as a video adapter 174. In addition to monitor 172, personal computers typically include other peripheral output devices (not shown), such as speakers and printers, which may be connected through output peripheral interface 175.

Computer 130 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 182. Remote computer 182 may include many or all of the elements and features described herein relative to computer 130.

Logical connections shown in Fig. 1 are a local area network (LAN) 177 and a general wide area network (WAN) 179. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, computer 130 is connected to LAN 177 via network interface or adapter 186. When used in a WAN networking environment, the computer typically includes a modem 178 or other means for establishing communications over WAN 179. Modem 178, which may be internal or external, may be connected to system bus 136 via the user input interface 170 or other appropriate mechanism.

Depicted in Fig. 1, is a specific implementation of a WAN via the Internet. Here, computer 130 employs modem 178 to establish communications with at least one remote computer 182 via the Internet 180.

In a networked environment, program modules depicted relative to computer 130, or portions thereof, may be stored in a remote memory storage device. Thus, e.g., as depicted in Fig. 1, remote application programs 189 may reside on a memory device of remote computer 182. It will be appreciated that the network connections shown and described are exemplary and other means of establishing a communications link between the computers may be used.

Attention is now drawn to Fig. 2, which is a block diagram depicting another exemplary device 200 that is also capable of benefiting from the methods and apparatuses disclosed herein. Device 200 is representative of any one or more devices or appliances that are operatively configured to process video and/or any related types of data in accordance with all or part of the methods and apparatuses described herein and their equivalents. Thus, device 200 may take the form of a computing device as in Fig.1, or some other form, such as, for example, a wireless device, a portable communication device, a personal digital assistant, a video player, a television, a DVD player, a CD player, a karaoke machine, a kiosk, a digital video projector, a flat panel video display mechanism, a set-top box, a video game machine, etc. In this example, device 200 includes logic 202 configured to process video data, a video data source 204 configured to provide video data to logic 202, and at least one display module 206 capable of displaying at least a portion of the video data for a user to view. Logic 202 is representative of hardware, firmware, software and/or any combination thereof. In certain implementations, for example, logic 202 includes a compressor/decompressor (codec), or the like. Video data source 204 is representative of any mechanism that can provide, communicate, output, and/or at least momentarily store video data suitable for processing by logic 202. Video reproduction source is illustratively shown as being within and/or without device 200. Display module 206 is representative of any mechanism that a user might view directly or indirectly and see the visual results of video data presented thereon. Additionally, in certain implementations, device 200 may also include some form or capability for reproducing or otherwise handling audio data associated with the video data. Thus, an audio reproduction module 208 is shown.

With the examples of Fig. 1 and Fig. 2 in mind, and others like them, the next sections focus on certain exemplary methods and apparatuses that may be at least partially practiced using with such environments and with such devices.

Conventional Direct Mode coding typically considerably improves coding efficiency of B frames by exploiting the statistical dependence that may exist between video frames. For example, Direct Mode can effectively represent block motion without having to transmit motion information. The statistical dependence that has been exploited thus far has been temporal dependence, which unfortunately implies that the timestamp information for each frame has to be available in both the encoder and decoder logic. Furthermore, the performance of this mode tends to deteriorate as the distance between frames increases since temporal statistical dependence also decreases. Such problems become even greater when multiple frame referencing is enabled, for example, as is the case of the H.264/AVC codec.

In this description improved methods and apparatuses are presented for calculating direct mode parameters that can achieve significantly improved coding efficiency when compared to current techniques. The improved methods and apparatuses also address the timestamp independency issue, for example, as described above. The improved methods and apparatuses herein build upon concepts that have been successfully adopted in P frames, such as, for example, for the encoding of a skip mode and exploiting the Motion Vector Predictor used for the encoding of motion parameters within the calculation of the motion information of the direct mode. An adaptive technique that efficiently combines temporal and spatial calculations of the motion parameters has been separately proposed.

In accordance with certain aspects of the present invention, the improved methods and apparatuses represent modifications, except for the case of the adaptive method, that do not require a change in the draft H.264/AVC bitstream syntax as it existed prior to July of 2002, for example. As such, in certain implementations the encoder and decoder region prediction logic may be the only aspects in such a standards-based system that need to be altered to support the improvements in compression performance that are described herein.

In terms of the use of these principles in a coding scheme such as the draft H.264/AVC standard, for example, other possible exemplary advantages provided by the improved methods and apparatuses include: timestamp independent calculation of direct parameters; likely no syntax changes; no extensive increase in complexity in the encoder logic and/or decoder logic; likely no requirement for (time-consuming/processor intensive) division in the calculations; considerable reduction of memory needed for storing motion parameters; relatively few software changes (e.g., when the motion vector prediction for 16x16 mode is reused); the overall compression-capability performance should be very close or considerably better than the direct mode in the H.264/AVC standard (software) as it existed prior to July of 2002; and enhanced robustness to unconventional temporal relationships with reference pictures since temporal relationship assumptions (e.g., such as assumptions that one reference picture for the coding of a B picture is temporally preceding the B picture and that the other reference picture for the coding of a B picture is temporally following the B picture) can be avoided in the MV prediction process.

In accordance with certain other aspects of the present invention, improvements on the current Rate Distortion Optimization (RDO) for B frames are also described herein, for example, by conditionally considering the Non-Residual Direct Mode during the encoding process, and/or by also modifying the Lagrangian λ parameter of the RDO. Such aspects of the present invention can be selectively combined with the improved techniques for Direct Mode to provide considerable improvements versus the existing techniques/systems.

Attention is drawn now to Fig. 3, which is an illustrative diagram depicting Direct Prediction in B frame coding, in accordance with certain exemplary implementations of the present invention.

The introduction of the Direct Prediction mode for a Macroblock/block within B frames, for example, is one of the main reasons why B frames can achieve higher coding efficiency, in most cases, compared to P frames. According to this mode as in the draft H.264/AVC standard, no motion information is required to be transmitted for a Direct Coded Macroblock/block, since it can be directly derived from previously transmitted information. This eliminates the high overhead that motion information can require. Furthermore, the direct mode exploits bidirectional prediction which allows for further increase in coding efficiency. In the example shown in Fig. 3, a B frame picture is coded with use of two reference pictures, a backward reference picture that is a P frame at a time t+2 that is temporally subsequent to the time t+1 of the B frame and a forward reference picture that is a P frame at a time t that is temporally previous to the B frame. It shall be appreciated by those familiar with the art that the situation shown in Fig. 3 is only an example, and in particular that the terms "forward" and "backward" may be used to apply to reference pictures that have any temporal relationship with the picture being coded (i.e., that a "backward" or "forward" reference picture may be temporally prior to or temporally subsequent to the picture being coded).

Motion information for the Direct Mode as in the draft H.264/AVC standard as it existed prior to July 2002 is derived by considering and temporally scaling the motion parameters of the collocated macroblock/block of the backward reference picture as illustrated in Fig. 3. Here, an assumption is made that an object captured in the video picture is moving with constant speed. This assumption makes it possible to predict a current position inside a B picture without having to transmit any motion vectors. By way of example, the motion vectors $(\overrightarrow{MV}_{fw}, \overrightarrow{MV}_{bw})$ of the Direct Mode versus the motion vector $\overrightarrow{MV}$ of the collocated block in the first backward reference frame can be calculated by:

$$\overrightarrow{MV}_{fw} = \frac{TR_B}{TR_D} \times \overrightarrow{MV} \text{ and } \overrightarrow{MV}_{bw} = \frac{(TR_B - TR_D)}{TR_D} \times \overrightarrow{MV}, \qquad (1)$$

where TR_B is the temporal distance between the current B frame and the reference frame pointed by the forward MV of the collocated MB, and TR_D is the temporal distance between the backward reference frame and the reference frame pointed by the forward MV of the collocated region in the backward reference frame. The same reference frame that was used by the collocated block was also used by the Direct Mode block. Until recently, for example, this was also the method followed within the work on the draft H.264/AVC standard, and still existed within the latest H.264/AVC reference software prior to July of 2002 (see, e.g., H.264/AVC Reference Software, unofficial software release Version 3.7).

As demonstrated by the example in Fig. 3 and the scaling equation (1) above, the draft H.264/AVC standard as it existed prior to July of 2002 and other like coding methods present certain drawbacks since they usually require that both the encoder and decoder have a priori knowledge of the timestamp information for each picture. In general, and especially due to the design of H.264/AVC which allows reference pictures almost anywhere in time, timestamps cannot be assumed by the order that a picture arrives at the decoder. Current designs typically do not include precise enough timing information in the syntax to solve this problem. A relatively new scheme was also under investigation for work on H.264/AVC, however, which in a sense does not require the knowledge of time. Here, the new H.264/AVC scheme includes three new parameters, namely, *direct_mv_scale_fwd*, *direct_mv_scale_bwd*, and *direct_mv_divisor* to the picture header and according to which the motion vectors of the direct mode can be calculated as follows:

$$\begin{aligned}\overrightarrow{MV}_{fw} &= \frac{direct\_mv\_scale\_fwd}{direct\_mv\_divisor}\times\overrightarrow{MV}\\ \overrightarrow{MV}_{bw} &= \frac{direct\_mv\_scale\_bwd}{direct\_mv\_divisor}\times\overrightarrow{MV}\end{aligned}\qquad(2)$$

Reference is now made to Fig. 4, which is an illustrative diagram depicting handling of collocated Intra within existing codecs wherein motion is assumed to be zero, in accordance with certain exemplary implementations of the present invention.

Reference is made next to Fig. 5, which is an illustrative diagram demonstrating that Direct Mode parameters need to be determined when the reference frame used to code the collocated block in the backward reference P picture is not the most recent reference picture that precedes the B picture to be coded (e.g., when the reference index is not equal to zero if the value zero for a reference index indicates the most recent temporally-previous forward reference picture). The new H.264/AVC scheme described above unfortunately has itself several drawbacks. For example, the H.264/AVC standard allows for multiple frame referencing and long-term storage of pictures as illustrated in Fig. 5. The new H.264/AVC scheme above fails to consider that different reference frames require different scaling factors. As such, for example, a significant reduction in coding efficiency has been reported (e.g., up to 10% loss in B frame coding efficiency). It is also quite uncertain what exactly the temporal relationship might be between the current block and its collocated block in such a case since the constant motion assumption described above is no longer followed. Additionally, temporal statistical relationships are reduced even further as reference frames become more temporally distant compared to one another.

Other issues include the inefficiency of the above new H.264/AVC scheme to handle intra blocks as shown in Fig. 4, for example, and/or even intra pictures. such as, for example, in the case of a scene change as shown in Fig. 6, which is an illustrative diagram showing a scene change and/or the situation wherein the collocated block is intra-coded. Currently, for example, a typically codec would assume that motion information is zero and use the first backward and forward reference pictures to perform bidirectional motion compensation. In this example, it may be more likely that the two collocated blocks from the forward and backward references have little, if any, relationship. Therefore, the usage of intra coding in the backward reference picture (shown as picture I in Fig. 6) in this case would likely cause a significantly reduction in the coding efficiency for the coding of the B pictures neighboring the scene change.

In the case of a scene change, for example, as in Fig. 6, where there is obviously no relationship between the two reference frames, a bidirectional prediction would usually provide no benefit. This implies that the Direct Mode, as previously defined, could be completely wasted. Unfortunately, current implementations of the Direct Mode usually are defined to always perform bidirectional prediction of a Macroblock/block.

Even if temporal distance parameters were available, it is not certain that the usage of the Direct Mode as conventionally defined is the most appropriate solution. In particular, for B frames that are temporally closer to a first temporally-previous forward reference frame, the statistical dependence might be much stronger with that frame than it would be for a temporally-subsequent backward reference frame. One example is a sequence where scene *A* changes to scene *B*, and then moves back to scene *A* (e.g., as might be the case in a news bulletin). The resulting performance of B frame encoding would likely suffer since Direct Mode will not be effectively exploited within the encoding process.

Unlike the conventional definitions of the Direct Mode where only temporal prediction was used, in co-pending Patent Application No. 10/444,511, which is incorporated herein by reference, several alternative improved methods and apparatuses are described for the assignment of the Direct Mode motion parameters wherein both temporal and/or spatial prediction are considered.

With these schemes and concepts in mind, in accordance with certain aspects of the present invention, presented below are some exemplary adaptive methods and apparatuses that combine such schemes and/or improve upon them to achieve even better coding performance under various conditions.

By way of example, in certain methods and apparatuses described below a high degree of statistical dependence of the motion parameters of adjacent macroblocks is exploited in order to further improve the efficiency of the SKIP Macroblock Mode for P pictures. For example, efficiency can be increased by allowing the SKIP mode to also use motion parameters, taken as the Motion Vector Predictor parameters of a current (16×16) Inter Mode. The same technique may also apply for B frames, wherein one may also generate both backward and forward motion vectors for the Direct mode using the Motion Vector Predictor of the backward or forward (16×16) Inter modes, respectively. It is also noted, for example, that one may even refine this prediction to other levels (e.g., 8×8, 4×4, etc.), however doing so would typically complicate the design.

In accordance with certain exemplary implementations of the present invention methods and apparatuses are provided to correct at least some of the issues presented above, such as, for example, the case of the collocated region in the backward reference picture using a different reference frame than the current picture will use and/or being intra coded. In accordance with certain other exemplary implementations of the present invention methods and apparatuses are provided which use a spatial-prediction based Motion Vector Predictor (MVP) concept to provide other benefits to the direct mode, such as, for example, the removal of division processing and/or memory reduction.

Direct Mode with INTRA and non-zero reference correction:

Fig. 7 is an illustrative diagram depicting a scheme wherein MV_{FW} and MV_{BW} are derived from spatial prediction (e.g., Median MV of forward and/or backward motion vector values of surrounding Macroblocks that use the same reference index) and wherein if either one is not available (e.g., no predictors) then one-direction prediction (e.g., forward-only prediction or backward-only prediction) may be used, in accordance with certain exemplary implementations of the present invention.

In accordance with certain exemplary methods, if a collocated block in the backward reference picture uses a zero-reference frame index and if also its reference picture exists in the reference buffer for the decoding process of the current picture to be decoded, then a scheme, such as, demonstrated above using equation (2) or the like is followed. Otherwise, a spatial-prediction based Motion Vector Predictor (MVP) for both directions (forward and backward) is used instead. By way of example, in the case of a collocated block being intra-coded or having a different reference frame index than the reference frame index to be used for the block of the current picture, or even the reference frame not being available anymore, then spatial-prediction MVP is used.

The spatial-prediction MVP can be taken, for example, as the motion vector predicted for the encoding of the current (16×16) Inter Mode (e.g., essentially with the usage of MEDIAN prediction or the like). This method in certain implementations is further modified by using different sized block or portions. For example, the method can be refined by using smaller block sizes. However, this tends to complicate the design sometimes without as much compression gain improvement. For the case of a Direct sub-partition within a P8x8 structure, for example, this method may still use a 16×16 MVD, even though this could be corrected to consider surrounding blocks.

Unlike the case of Skip Mode in a P picture, in accordance with certain aspects of the present invention, the motion vector predictor is not restricted to use exclusively the zero reference frame index. Here, for example, an additional Reference Frame Prediction process may be introduced for selecting the reference frame that is to be used for either the forward or backward reference. Those skilled in the art will recognize that this type of prediction may also be applied in P frames as well.

If no reference exists for prediction (e.g., the surrounding Macroblocks are using forward prediction and thus there exists no backward reference), then the direct mode can be designed such that it becomes a single direction prediction mode. This consideration can potentially solve several issues such as inefficiency of the H.264/AVC scheme prior to July of 2002 in scene changes, when new objects appear within a scene, etc. This method also solves the problem of both forward and backward reference indexes pointing to temporally-future reference pictures or both pointing to temporally-subsequent reference pictures, and/or even when these two reference pictures are the same picture altogether.

For example, attention is drawn to Fig. 8, which is an illustrative diagram depicting how spatial prediction may be employed to solve the problem of scene changes and/or that Direct Mode need not be restricted to being Bidirectional, in accordance with certain exemplary implementations of the present invention. Here, as described above and illustrated, the Direct Mode need not necessarily be bidirectional.

Presented below is exemplary pseudocode for such a method. In this pseudo-code, it is assumed that the value -1 is used for a reference index to indicate a non-valid index (such as the reference index of an intra region) and it is assumed that all values of reference index are less than 15, and it is assumed that the result of an "&" operation applied between the number -1 and the number 15 is equal to 15 (as is customary in the C programming language). It is further assumed that a function SpatialPredictor(Bsize,X,IndexVal) is defined to provide a motion vector prediction for a block size Bsize for use in a prediction of type X (where X is either FW, indicating forward prediction or BW, indicating backward prediction) for a reference picture index value IndexVal. It is further assumed that a function min(a,b,c) is defined to provide the minimum of its arguments a, b, and c. It is further assumed for the purpose of this example that the index value 0 represents the index of the most commonly-used or most temporally closest reference picture in the forward or backward reference picture list, with increasing values of index being used for less commonly-used or temporally more distant reference pictures.

```
            Direct_MV_Calculation()
            {
             if (CollocatedRegionRefIndex!=0)
             {
              // Note that UpRight can be replaced by UpLeft at frame
    boundaries

    FwReferenceIndex=min(referenceBfwLeft&15,referenceBfwUp&15,refere
    nceBfwUpRight&15);

    BwReferenceIndex=min(referenceBbwLeft&15,referenceBbwUp&15,refer
    enceBbwUpRight&15)
               if FwReferenceIndex!=15
               {
                DirectMVfw=SpatialPredictor(16x16,FW,FwReferenceIndex);
                referenceIndexBfw=FwReferenceIndex;
               }
               else
               {
                DirectMVfw=0;
                referenceIndexBfw= -1;
               }
            if BwReferenceIndex!=15
            {
             DirectMVbw=SpatialPredictor(16x16,BW,BwReferenceIndex);
             referenceIndexBbw=BwReferenceIndex;
            }
            else
            {
             DirectMVbw=0;
             referenceIndexBbw= -1;
            }
            if (BwReferenceIndex==15 && FwReferenceIndex==15)
             referenceIndexBbw=referenceIndexBfw=0;
         }
         else // Perform Prediction using temporal information
         {

   DirectMVfw=direct_mv_scale_fwd*MvP/direct_mv_scale_divisor;

   DirectMVbw=direct_mv_scale_bwd*MvP/direct_mv_scale_divisor;
          referenceIndexBfw=0;
          referenceIndexBbw=0;
         }
        }
```

In the above algorithm, if the collocated block in the backward reference picture uses the zero-index reference frame (e.g., CollocatedRegionRefIndex == 0), and the temporal prediction MVs are calculated for both backward and forward prediction as the equation (2); otherwise the spatial MV prediction is used instead. For example, the spatial MV predictor first examines the reference indexes used for the left, up-left and up-right neighboring macroblocks and finds the minimum index value used both forward and backward indexing. If, for example, the minimum reference index is not equal to 15 (Fw/BwReferenceIndex = 15 means that all neighboring macroblocks are coded with Intra), the MV prediction is calculated from spatial neighboring macroblocks. If the minimum reference index is equal to 15, then the MV prediction is zero.

The above method may also be extended to interlaced frames and in particular to clarify the case wherein a backward reference picture is coded in field mode, and a current picture is coded in frame mode. In such a case, if the two fields have different motion or reference frame, they complicate the design of direct mode with the original description. Even though averaging between fields could be applied, the usage of the MVP immediately solves this problem since there is no dependency on the frame type of other frames. Exceptions in this case might include, however, the case where both fields have the same reference frame and motion information.

In addition, in the new H.264/AVC standard the B frame does not constrain its two references to be one from a previous frame and one from a subsequent frame. As shown in the illustrative timeline in Fig. 23, both reference pictures (forward and backward, also known as List 0 and List 1) of a macroblock partition may precede a current picture in temporal order. The methods and apparatuses provided herein are also suitable for use in this case. Alternatively, both reference pictures may be temporally subsequent to a current picture. Thus, usage of MVP does not depend on the order of references.

Division Free, Timestamp independent Direct Mode:

In the exemplary method above, the usage of the spatial-prediction based MVP for some specific cases solves various prediction problems in the current direct mode design. There still remain, however, several issues that are addressed in this section. For example, by examining equation (2) above, one observes that the calculation of the direct mode parameters requires a rather computationally expensive division process (for both horizontal and vertical motion vector components). This division process needs to be performed for every Direct Coded subblock. Even with the improvements in processing technology, division tends to be a highly undesirable operation, and while shifting techniques can help it is usually more desirable to remove as much use of the division calculation process as possible.

Furthermore, the computation above also requires that the entire motion field (including reference frame indexes) of the first backward reference picture be stored in both the encoder and decoder. Considering, for example, that blocks in H.264/AVC may be of 4×4 size, storing this amount of information may become relatively expensive as well.

With such concerns in mind, attention is drawn to Fig. 9, which is a flow diagram depicting an exemplary method 900 for Timestamp Independent SpatioTemporal Prediction for Direct Mode, in accordance with certain exemplary implementations of the present invention. Here, in act 902, spatial predictors MV_a, MV_b, and MV_c are provided/determined along with temporal predictor MV_t. In act 904, MV_{Direct} is determined, in this example, as the Median of MV_a, MV_b, and MV_c. In act 906 it is determined if MV_t is zero, and if so, then method 900 continues with act 908, otherwise method 900 continues with act 910. In act 908, MV_{Direct} is set to zero and the method ends with this being the output. In act 910, it is determined if $MV_a=0 \; || \; MV_b=0 \; || \; MV_c=0$, if so, then according to act 908 MV_{Direct} is set to zero and the method ends with this being the output, otherwise, then MV_{Direct} remains as set in act 904 and the method ends with this being the output.

Those skilled in the art will recognize that other suitable linear and/or non-linear functions may be substituted for the exemplary Median function in act 904.

The usage of the spatial-prediction based MVP though does not require any such operation or memory storage. Thus, it is recognized that using the spatial-prediction based MVP for all cases, regardless of the motion information in the collocated block of the first backward reference picture may reduce if not eliminate many of these issues.

Even though one may disregard motion information from the collocated block, in the present invention it was found that higher efficiency is usually achieved by also considering whether the collocated block is stationary and/or better, close to stationary. In this case motion information for the direct mode may also be considered to be zero as well. Only the directions that exist, for example, according to the Reference Frame Prediction, need be used. This concept tends to protect stationary backgrounds, which, in particular at the edges of moving objects, might become distorted if these conditions are not introduced. Storing this information requires much less memory since for each block only 1 bit needs to be stored (to indicate zero/near-zero vs. non-zero motion for the block).

By way of further demonstration of such exemplary techniques, the following pseudocode is presented:

```
Direct_MV_Calculation()
{
 // Note that UpRight can be replaced by UpLeft at frame boundaries

FwReferenceIndex=min(referenceBfwLeft&15,referenceBfwUp&15,referenceBfwUpRight&15);

BwReferenceIndex=min(referenceBbwLeft&15,referenceBbwUp&15,referenceBbwUpRight&15)
    if FwReferenceIndex!=15
    {
```

```
                 if (!CollocatedRegionRefIndex && (!(abs(MvPx)>>1))&&
(!(abs(MvPy)>>1)) // Examine if stationary collocated
                {
                 DirectMVfw=0;
                 referenceIndexBfw=0;
                }
                else
                {
                 DirectMVfw=SpatialPredictor(16x16,FW,FwReferenceIndex);
                 referenceIndexBfw=FwReferenceIndex;
                }
            }
            else
            {
             DirectMVfw=0;
             referenceIndexBfw=-1;
            }
            if BwReferenceIndex!=15
            {
                 if (!CollocatedRegionRefIndex && (!(abs(MvPx)>>1))&&
(!(abs(MvPy)>>1)) // Examine if stationary collocated
                {
                 DirectMVbw=0;
                 referenceIndexBbw=0;
                }
                else
                {

DirectMVbw=SpatialPredictor(16x16,BW,BwReferenceIndex);
                 referenceIndexBbw=BwReferenceIndex;
                }
            }
            else
            {
             DirectMVbw=0;
             referenceIndexBbw=-1;
            }
            if (BwReferenceIndex==15 && FwReferenceIndex==15)
             referenceIndexBbw=referenceIndexBfw=0;
        }
```

In the above, the MV predictor directly examines the references of neighboring blocks and finds the minimum reference in both the forward and backward reference picture lists. Then, the same process is performed for the selected forward and backward reference index. If, for example, the minimum reference index is equal to 15, e.g., all neighboring blocks are coded with Intra, the MV prediction is zero. Otherwise, if the collocated block in the first backward reference picture uses a zero-reference frame index and has zero or very close to zero motion (e.g., MvPx = 0 or 1 or -1), the MV prediction is zero. In the rest of the cases, the MV prediction is calculated from spatial information.

This scheme performs considerably better than the H.264/AVC scheme as it existed prior to July of 2002 and others like it, especially when the distance between frames (e.g., either due to frame rate and/or number of B frames used) is large, and/or when there is significant motion within the sequence that does not follow the constant motion rules. This makes sense considering that temporal statistical dependence of the motion parameters becomes considerably smaller when distance between frames increases.

Adaptive Selection of Direct Mode type at the Frame level:

Considering that both of the above improved exemplary methods/schemes have different advantages in different types of sequences (or motion types), but also have other benefits (i.e., the second scheme requiring reduced division processing, little additional memory, storage/complexity), in accordance with certain further aspects of the present invention, a combination of both schemes is employed. In the following example of a combined scheme certain decisions are made at a frame/slice level.

According to this exemplary combined scheme, a parameter or the like is transmitted at a frame/slice level that describes which of the two schemes is to be used. The selection may be made, for example, by the user, an RDO scheme (e.g., similar to what is currently being done for field/frame adaptive), and/or even by an "automatic pre-analysis and pre-decision" scheme (e.g., see Figs 10a-b).

Figs 10a-b are illustrative diagrams showing how Direct/Skip Mode decision can be performed either by an adaptive frame level RDO decision and/or by user scheme selection, respectively, in accordance with certain exemplary implementations of the present invention.

In Fig. 10a method 1000 includes act 1002 wherein the input image is provided to a plurality of different Direct or Copy encoding schemes, herein illustrated by acts 1004 and 1006. Act 1004 employs direct scheme encoding, where the MV prediction is calculated from temporal information as the equation (2). Act 1006 employs copy scheme encoding, where the MV prediction is calculated from spatial information. Once the input image has been encoded per acts 1004 and 1006, then in act 1008, an RDO or other like decision is made to select a desired encoded image output.

In Fig. 10b method 1020 includes act 1002 wherein the input image is provided to act 1022 wherein a scheme decision is made to selectively determine which if any of a plurality of different Direct or Copy encoding schemes will be employed, herein illustrated by acts 1024 and 1026. The decision in act 1022 can be explicitly accomplished with user inputs, for example. In act 1022, more intelligent and/or automated methods may be used for more optimally selecting the best direct method for each picture. Act 1024 employs a direct scheme encoding, where the MV prediction is calculated from temporal information as the equation (2). Act 1026 employs a copy scheme encoding, where the MV prediction is calculated from spatial information. Once the input image has been encoded per selected acts 1024, 1026 and/or otherwise provided, then in act 1028, another selection decision is made to select a desired output.

In certain implementations, one of the schemes, such as, for example, scheme B (acts 1006 and 1026) is made as a mandatory scheme. This would enable even the simplest devices to have B frames, whereas scheme A (acts 1004 and 1024) could be an optional scheme which, for example, one may desire to employ for achieving higher performance.

Decoding logic/devices which do not support this improved scheme could easily drop these frames by recognizing them through the difference in syntax. A similar design could also work for P pictures where, for some applications (e.g., surveillance), one might not want to use the skip mode with Motion Vector Prediction, but instead use zero motion vectors. In such a case, the decoder complexity will be reduced.

An exemplary proposed syntax change within a slice header of the draft H.264/AVC standard is shown in the table listed in Fig. 11. Here, the new additional parameters are copy_mv_spatial and direct_mv_spatial for P pictures and B pictures respectively. Value 0 for these parameters implies Skip on MVP for P frames, and MVP Direct for B frames. If MVP Direct is used (direct_mv_spatial=0), it is not necessary to transmit the additional direct parameters.

A potential scenario in which the above design might give considerably better performance than the draft H.264/AVC scheme prior to July of 2002 can be seen in Fig. 12, which is an illustrative diagram depicting different frames signal different type of prediction for their corresponding Direct (B) and Skip (P) modes. Here, P_Z, P_T, and P_M, define for example zero, temporal and spatial prediction, and

B_T, B_{SP}, define temporal and spatial prediction for Direct Mode, in accordance with certain exemplary implementations of the present invention.

In certain implementations, instead of transmitting the direct_mv_scale_divisor parameter a second parameter direct_mv_scale_div_diff may be transmitted and which is equal to:

$$\begin{aligned}&direct\_mv\_scale\_div\_diff =\\&direct\_mv\_scale\_divisor -\\&(direct\_mv\_scale\_fwd - direct\_mv\_scale\_bwd).\end{aligned}$$

Exemplary Performance Analysis

Simulation results were performed according to the test conditions specified in G. Sullivan, "Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material", document VCEG-N81, Sep. 2001.

The performance was tested for both UVLC and CABAC entropy coding methods of H.264/AVC, with 1-5 reference frames, whereas for all CIF sequences we used 1/8th subpixel motion compensation. 2B frames in-between P frames were used. Some additional test sequences were also selected. Since it is also believed that bidirectional prediction for block sizes smaller than 8×8 may be unnecessary and could be quite costly to a decoder, also included are results for the MVP only case with this feature disabled. RDO was enabled in the experiments. Some simulation results where the Direct Mode parameters are calculated according to the text are also included, but without considering the overhead of the additional parameters transmitted.

Currently the RDO of the system uses the following equation for calculating the Lagrangian parameter λ for I and P frames:

$$\lambda_{I,P} = 0.85 \times 2^{\frac{QP}{3}} \qquad (3)$$

where *QP* is the quantizer used for the current Macroblock. The B frame λ though is equal to $\lambda_B = 4 \times \lambda_{I,P}$.

Considering that the usage of the MVP requires a more accurate motion field to work properly, it appears from this equation that the λ parameter used for B frames might be too large and therefore inappropriate for the improved schemes presented here.

From experiments it has been found that an adaptive weighting such as:

$$f(QP) = \max\left(2, \min(4, \frac{QP}{6})\right)$$

tends to perform much better for the *QP* range of interest (e.g., $QP \in \{16, 20, 24, 28\}$). In this exemplary empirical formula QP/6 is truncated between 2 and 4 because λ_B has no linear relationship with $\lambda_{I,P}$ when QP is too large or too small. Furthermore, also added to this scheme was a conditional consideration of the Non-Residual Direct mode since, due to the (16×16) size of the Direct Mode, some coefficients might not be completely thrown away, whereas the non residual Direct mode could improve efficiency.

It was also found that the conditional consideration, which was basically an evaluation of the significance of the Residual Direct mode's Coded Block Pattern (CBP) using $MOD(CBP,16) < 5$, behaves much better in the RDO sense than a non conditional one. More particularly, considering that forcing a Non-RDO mode essentially implies an unknown higher quantization value, the performance of an in-loop de-blocking filter deteriorates. The error also added by this may be more significant than expected especially since there can be cases wherein no bits are required for the encoding of the NR-Direct mode, thus not properly using the λ parameter. In addition, it was also observed that using a larger quantizer such as $QP + N$ ($N > 0$) for B frames would give considerably better performance than the non conditional NR-Direct consideration, but not compared to the conditional one.

The experimental results show that the usage of the MVP, apart from having several additional benefits and solving almost all, if not all, related problems of Direct Mode, with proper RDO could achieve similar if not better performance than conventional systems.

The performance of such improved systems is dependent on the design of the motion vector and mode decision. It could be argued that the tested scheme, with the current RDO, in most of the cases tested is not as good as the partial MVP consideration with the same RDO enabled, but the benefits discussed above are too significant to be ignored. It is also pointed out that performance tends to improve further when the distance between the reference images increases. Experiments on additional sequences and conditions (including 3B frames) are also included in the table shown in Fig. 21.

As such, Fig. 21 is a table showing the performance difference of exemplary proposed schemes and proposed RDO versus conventional software (i.e., H.264/AVC JM3.3), in accordance with certain exemplary implementations of the present invention.

Here, the resulting performance of the improved scheme versus previously reported performance may be due at least in part to the larger λ of the JM version that was used, which basically benefited the zero reference more than others. Finally, not using block sizes smaller than 8×8 for bidirectional prediction does not appear to have any negative impact in the performance of the improved scheme/design.

Fig. 13 is a table showing modifications to modes for 8x8 blocks in B pictures/slices applicable to the draft H.264/AVC scheme, in accordance with certain exemplary implementations of the present invention. Experimental results show that the addition of this improvement reduces the efficiency of the improved scheme by only about 0.5% on the average (i.e., about 0.02dB).

It is also noted that for different sequences of frames the two proposed schemes (e.g., A, B) demonstrate different behavior. It appears that the adaptive selection tends to improve performance further since it makes possible the selection of the better/best possible coding scheme for each frame/slice. Doing so also enables lower capability devices to decode MVP only B frames while rejecting the rest.

Motion Vector (MV) prediction will now be described in greater detail based on the exemplary improved schemes presented herein and the experimental results and/or expectations associated there with.

Motion Vector Prediction description:

Fig. 14 is an illustrative diagram depicting median prediction of motion vectors, in accordance with certain exemplary implementations of the present invention.

The draft H.264/AVC scheme is obscure with regards to Motion Vector Prediction for many cases. According to the text, the vector component E of the indicated block in Fig. 14 is predicted normally as the median of A, B and C. However, the prediction may be modified as described below:

A The component applying to the sample to the left of the upper left sample in E

B The component applying to the sample just above the upper left sample in E

C The component applying to the sample above and to the right of the upper right sample in E

D The component applying to the sample above and to the left of the upper left sample in E

A, B, C, D and E may represent motion vectors from different reference pictures. The following substitutions may be made prior to median filtering:

Rule 1: If A and D are outside the picture, their values are assumed to be zero and they are considered to have "different reference picture than E".

Rule 2: If D, B, and C are outside the picture, the prediction is equal to A (equivalent to replacing B and C with A before median filtering).

Rule 3: If C is outside the picture or still not available due to the order of vector data (see Figure 2), C is replaced by D.

If any of the blocks A, B, C, D are intra coded then they count as having a "different reference picture". If one and only one of the vector components used in the median calculation (A, B, C) refer to the same reference picture as the vector component E, this one vector component is used to predict E.

By examining, all possible combinations according to the above, the table in Fig. 15 can be generated. Here, for example, Fig. 15 shows a table for P-Picture

Motion Vector prediction (e.g., Non-Skip, non-8x16, non-16x8 MBs), in accordance with certain exemplary implementations of the present invention.

In this context, "availability" is determined by whether a macroblock is "outside the picture" (which is defined to include being outside the slice as well as outside the picture) or "still not available due to the order of vector data". According also to the above text, if a block is available but intra, a macroblock A, B, C, or D is counted as having a "different reference picture" from E, but the text does not specify what motion vector value is used. Even though the software assumes this is zero, this is not clearly described in the text. All these cases and rules can also be illustrated by considering Fig. 16, which is an illustrative diagram depicting median prediction of motion vectors, in accordance with certain exemplary implementations of the present invention.

To solve the above issues and clarify completely motion vector prediction, it is proposed that the following exemplary "rule changes" be implemented in such a system according to which the main difference is in modifying Rule 1 (above) and merging it with Rule 4, for example, as listed below:

Rule 0: Median rule is applied for Motion vector calculation:

M_E=Median (M_A,M_B,M_C)

Rule 1: If a predictor is outside of the picture/slice or is intra then this predictor is assumed to have zero motion vectors and "different reference picture than E".

Rule 2: If B & C & D outside of picture $\Rightarrow M_E = M_A$, i.e., if D, B, and C are outside the picture, the prediction of E is equal to A

Rule 3: If C not available (outside of picture, not yet coded etc) C is replaced by D

Rule 4: If x ($x \in$ A, B ,C) and only x has $R_x == R_E$ then $M_E = M_x$

The interpretation of Rule 4 is, if only one (referred to x) of A, B, C has the same MV as E in the reference frame, and then the prediction of E is equal to x.

These exemplary modified rules are adaptable for H.264/AVC, MPEG or any other like standard or coding logic process, method, and/or apparatus.

Fig. 17, for example, is an illustrative diagram showing replacement of Intra subblock predictors with adjacent Inter subblock predictors, in accordance with certain exemplary implementations of the present invention and the above exemplary modified rules.

Some additional exemplary rules that may also be implemented and which provide some further benefit in encoding include:

Rule W: If x_1 ($x_1 \in$ A, B ,C) and x_2 ($x_2 \in$ A, B ,C, $x_2 \neq x_1$) are intra and x_3 ($x_3 \in$ A, B ,C, $x_3 \neq x_2 \neq x_1$) is not, then only x_3 is used in the prediction.

The interpretation of rule W is, if two of A, B, C are coded with Intra and the third is coded with Inter, and then it is used in the prediction.

Rule X: Replacement of intra subblock predictors (due to tree structure) by adjacent non intra subblock within same Macroblock for candidates A and B (applicable only to 16×16, 16×8, and 8×16 blocks), e.g., as in Fig. 17.

Rule Y: If TR information is available, motion vectors are scaled according to their temporal distances versus the current reference. See, for example, Fig. 18, which is an illustrative diagram depicting how Motion Vector Prediction of current block (C) may consider the reference frame information of the predictor macroblocks (Pr) and perform the proper adjustments (e.g., scaling of the predictors), in accordance with certain exemplary implementations of the present invention.

With Rule Y, if predictors A, B, and C use reference frames RefA, RefB, and RefC, respectively, and the current reference frame is Ref, then the median predictor is calculated as follows:

$$\overrightarrow{MV}_{pred} = Ref \times Median\left(\frac{\overrightarrow{MV}_A}{RefA}, \frac{\overrightarrow{MV}_B}{RefB}, \frac{\overrightarrow{MV}_C}{RefC}\right) \qquad (4)$$

It has been found that computation such as this can significantly improve coding efficiency (e.g., up to at least 10% for P pictures) especially for highly temporally consistent sequences such as sequence Bus or Mobile. Considering Direct Mode, TR, and division, unfortunately, even though performance-wise such a solution sounds attractive, it may not be suitable in some implementations.

Rule Z: Switching of predictor positions within a Macroblock (e.g., for left predictor for the 16x16 Mode), use the A1 instead of A2 and B2 instead of B1 as shown, for example, in Fig. 19, which is an illustrative diagram depicting certain exemplary predictors for 8×8 partitioning, in accordance with certain exemplary implementations of the present invention.

Performance Analysis of Lagrangian Parameter Selection:

Rate Distortion Optimization (RDO) with the usage of Lagrangian Parameters (λ) represent one technique that can potentially increase coding efficiency of video coding systems. Such methods, for example, are based on the principle of jointly minimizing both Distortion D and Rate R using an equation of the form:

$$J = D + \lambda \cdot R \qquad (5)$$

The JVT reference encoding method for the draft H.264/AVC standard as it existed prior to July of 2002, for example, has adopted RDO as the encoding method of choice, even though this is not considered as normative, whereas all testing conditions of new proposals and evaluations appear to be based on such methods.

The success of the encoding system appears highly dependent on the selection of λ which is in the current software selected, for I and P frame, as:

$$\lambda_{I,P} = 0.85 \times 2^{\frac{QP}{3}}$$

where QP is the quantizer used for the current Macroblock, and

$$\lambda_B = 4 \times \lambda_{I,P}$$

is used for B frames.

In accordance with certain aspects of the present invention, it was determined that these functions can be improved upon. In the sections below, exemplary analysis into the performance mainly with regard to B frames is provided. Also, proposed is an improved interim value for λ.

Rate Distortion Optimization:

By way of example, the H.264/AVC reference software as it existed prior to July of 2002 included two different complexity modes used for the encoding of a sequence, namely, a high complexity mode and a lower complexity mode. As described above, the high complexity mode is based on a RDO scheme with the usage of Lagrangian parameters which try to optimize separately several aspects of the encoding. This includes motion estimation, intra block decision, subblock decision of the tree macroblock structure, and the final mode decision of a macroblock. This method depends highly on the values of λ which though have been changed several times in the past. For example, the value of λ has recently change from

$$\lambda_{I,P} = 5 \times \frac{QP+5}{34-QP} \times \exp^{\frac{QP}{10}} \qquad (6)$$

to

$$\lambda_{I,P} = 0.85 \times 2^{\frac{QP}{3}} \qquad (7)$$

or basically

$$\lambda_{I,P} = \frac{A}{1000} \times 2^{\frac{QP}{3}} \qquad (8)$$

where A=850, mainly since the previous function could not accommodate the new QP range adopted by the standard. Apparently though the decision of changing the value of λ appears to most likely have been solely based on P frame performance, and probably was not carefully tested.

In experiments conducted in the present invention discovery process it was determined that, especially for the testing conditions recommended by the JVT prior to July of 2002, the two equations are considerably different. Such a relationship can be seen in Fig. 20, which is a table showing the relationship between previous λ and current λ, in accordance with certain exemplary implementations of the present invention.

Here, one can note that for the range (16, 20, 24, 28) the new λ is, surprisingly, between 18% and 36% larger than the previous value. The increase in λ can have several negative effects in the overall performance of the encoder, such as in reduced reference frame quality and/or at the efficiency of motion estimation/prediction.

It is pointed out that the PSNR does not always imply a good visual quality, and that it was observed that in several cases several blocking artifacts may appear even at higher bit rates. This may also be affected by the usage of the Non-residual skip mode, which in a sense bypasses the specified quantizer value and thus reduces the efficiency of a deblocking filter. This may be more visually understood when taking in consideration that this mode could in several cases require even zero bits to be encoded, thus minimizing the effect of the λ (λ depends on the original *QP*). Considering that the distortion of all other, more efficient, macroblock modes is penalized by the larger value of λ it becomes apparent that quite possibly the actual coding efficiency of the current codec has been reduced. Furthermore, as mentioned above, the new value was most likely not tested within B frames.

In view of the fact that B frames rely even more on the quality of their references and use an even larger Lagrangian parameter ($\lambda_B = 4 \times \lambda_{I,P}$), experimental analysis was conducted to evaluate the performance of the current λ when B frames are enabled. Here, for example, a comparison was done regarding the performance with A=500 and A=700 (note that the later gives results very close to the previous, *e*-based λ).

In the experimental design, the λ for B frames was calculated as:

$$\lambda_B = \max\left(2, \min(4, \frac{QP}{6})\right) \times \lambda_{I,P}$$

since at times $\lambda_B = 4 \times \lambda_{I,P}$ was deemed to excessive. In this empirical formula QP/6 is truncated between 2 and 4 because λ_B has no linear relationship with $\lambda_{I,P}$ when QP is too large or too small.

Based on these experiments, it was observed that if the same *QP* is used for both B and P frames, *A*=500 outperforms considerably the current λ (*A*=850). More specifically, encoding performance can be up to about at least 2.75% bit savings (about 0.113dB higher) for the exemplary test sequences examined. The results are listed in Fig. 22, which is a table showing a comparison of encoding performance for different values of λ, in accordance with certain exemplary implementations of the present invention.

Considering the above performance it appears that an improved value of *A* between about 500 and about 700 may prove useful. Even though from the above results the value of 500 appears to give better performance in most cases (except container) this could affect the performance of P frames as well, thus a larger value may be a better choice. In certain implementations, for example, *A*=680 worked significantly well.

Conclusion

Although the description above uses language that is specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the invention.